

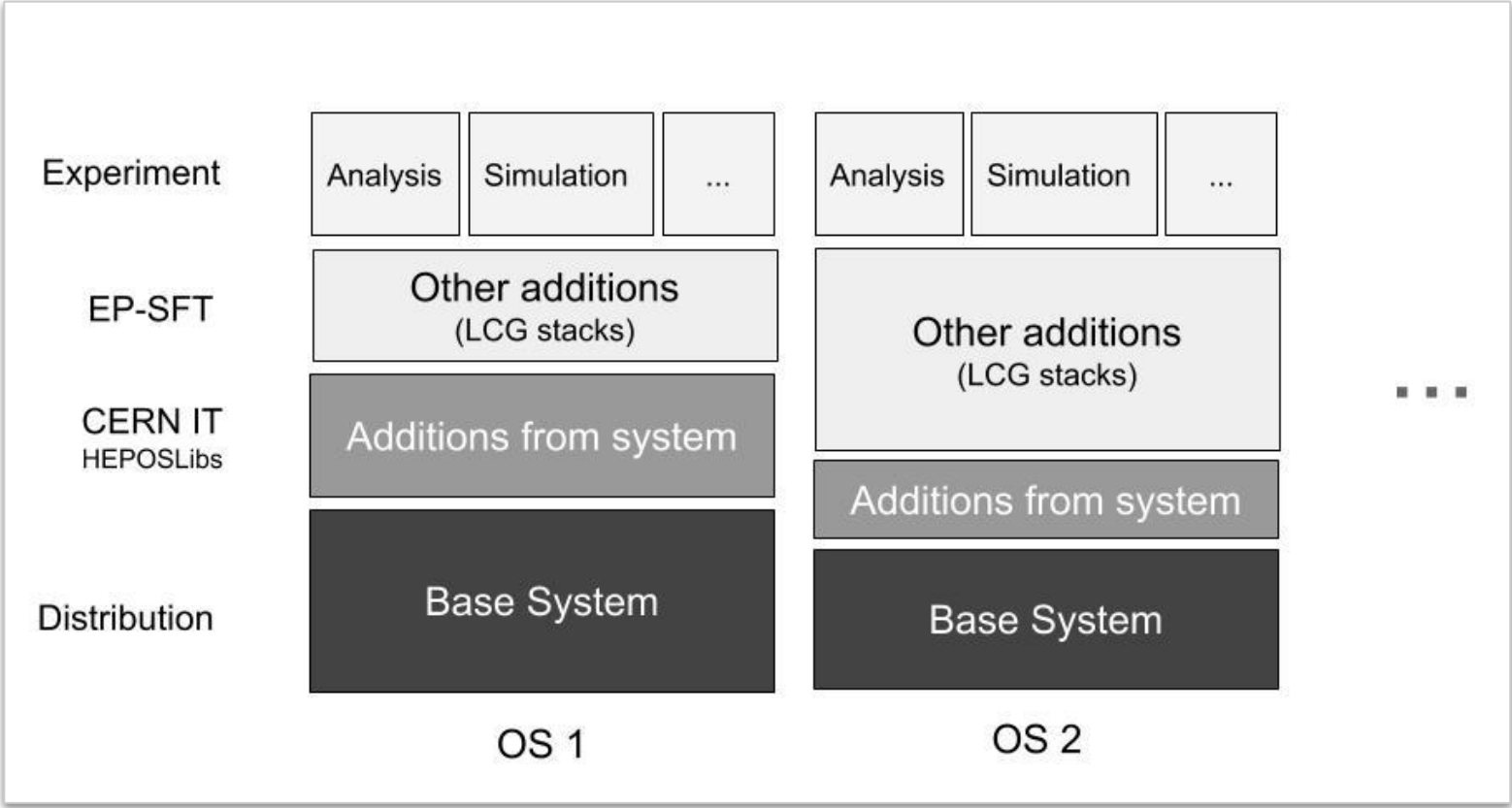
# The LCG Software Stacks

---

pre-GDB on  
Software Deployment

CERN virtual, 5 May 2020  
D Konstantinov (IHEP - NRC Kurchatov Institute)  
On behalf of the EP-SFT/SPI team

# The LCG Software Stacks in the overall picture



# Matching experiments/user needs

- Baseline CERN Linux distros favour long term stability

	released	CERN	RHEL	Native gcc	Python	EOL
SLC6	03/2011	~2014	6	4.4.7	2.6.6	11/2020
Centos7	07/2014	04/2019	7	4.8.5	2.7.5	2024
CentOS8	10/2019		8	8.3.1	2.7.16/3.6.8	2029

- Experiments / users need modern compilers and package version
  - Recent compiler improvements, recent package versions
  - Packages not provided by distributions (e.g. MC generators, in-house proj)
- LCG stacks role is to provide the match

# The LCG Software Stacks in a nutshell

- Contents

- ~450 packages
  - contrib: gcc, clang, ...
  - projects: ROOT, Geant4, COOL/CORAL, DD4hep, VecGeom, ...
  - externals: external, non-purely Python external packages
  - pyexternals: purely Python external packages
  - generators: physics generators

- Main platforms/compilers

- Baseline stable CERN linux distros: SLC6, CERN-CentOS7 (next: CentOS8)
  - Also Ubuntu LTS, MacOSX
- Latest compilers: gcc8, gcc9, clang10
- Baseline Python2, early view on Python3
- Debug builds

# Users

- LHC experiments: ATLAS, LHCb
- CERN experiments and projects: SWAN, NA62, BE Department, FCC
- Many individual HEP and not only users and projects around the world.

Reminder:

Bi-weekly “Librarians and Integrators Meetings”, so called LIM-meetings:

<https://indico.cern.ch/category/1449/>

CERN e-mailing list: **project-lcg-app-lim**

Special new mailing list for LCG stack users: **lcgstack-users**

**JIRA tickets: <https://sft.its.cern.ch/jira/projects/SPI/issues>**

# Building tools and infrastructure

LCG Stack is built using LCGCmake

- CMake-based set of tools
- Specific “stack” defined by a toolchain
- Recently improved to support layered stacks

Build infrastructure orchestrated with Jenkins

- Using docker containers in CERN OpenStack Centos7 VM build nodes
  - Modified CERN base docker image = CERN base + HepOSlibs + some others
- Content/rules/strategy for HepOSlibs discussed during [May 12th LIM](#)
  - See also next talk

# Deliverables, Packaging and Deployment

## Deliverables:

- Two main releases per year (roughly following main ROOT releases)
- Several nightlies deliverables
  - dev4 (ROOT-pro-patches, ...), dev3 (ROOT-master, ...), dev3python3, dev4cuda, ... and others (see next page)

## Packaging:

- Binary tarfiles + dependency meta-information (.txt files)
- Deployed on CernVM-FS: **`/cvmfs/sft.cern.ch`**, **`/cvmfs/sft-nightlies.cern.ch`**
- Repacked into RPMs (used by ATLAS, LHCb)

# “Exotic” LCG stack flavours

- ARM platforms: aarch64-centos7-gcc8-opt and aarch64-centos7-gcc9-opt built using ThunderX servers provided by OpenLab
  - [/cvmfs/sft-nightlies.cern.ch/lcg/nightlies/devARM/](https://cvmfs/sft-nightlies.cern.ch/lcg/nightlies/devARM/)
- CUDA-enabled builds with cuda 10.1
  - [/cvmfs/sft.cern.ch/lcg/releases/LCG\\_96py3cu10](https://cvmfs/sft.cern.ch/lcg/releases/LCG_96py3cu10)
  - [/cvmfs/sft.cern.ch/lcg/releases/LCG\\_97py3cu10](https://cvmfs/sft.cern.ch/lcg/releases/LCG_97py3cu10)



# LCG Views

- Simplify the setup of a given stack
- Contains **symlinks** to all what required to enable a given release/nightly stack
- Path with Linux-like organisation (bin, lib, ...) of packages and files under
  - /cvmfs/sft.cern.ch/lcg/views
  - /cvmfs/sft-nightlies.cern.ch/lcg/views
- Convenient setup script set all necessary environment variables:

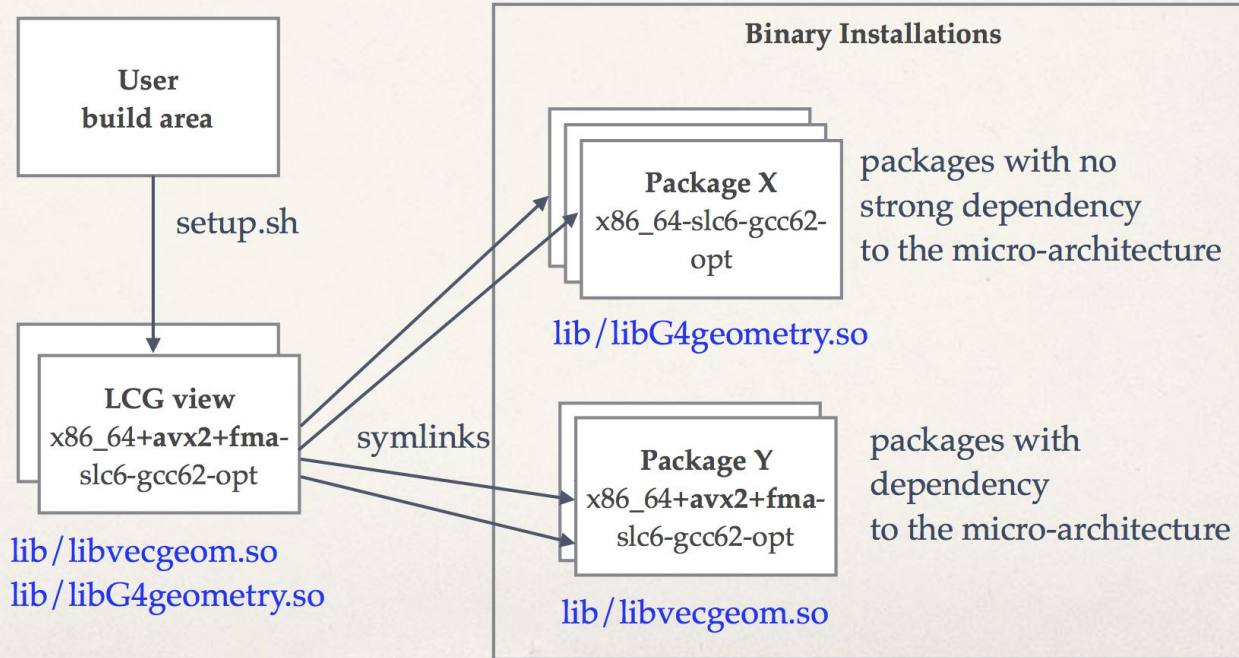
```
lxplus:~ $ source /cvmfs/sft.cern.ch/lcg/views/LCG_97/x86_64-centos7-gcc9-opt/setup.sh
lxplus:~ $ gcc -v
...
gcc version 9.2.0 (GCC)
lxplus:~ $ which root
/cvmfs/sft.cern.ch/lcg/views/LCG_97/x86_64-centos7-gcc9-opt/bin/root
```

# Support Multiple Micro-architectures

- Separate stacks for specific microarchitectures, supporting specific instructions sets (fma, avx, avx2, sse4,...), using appropriate flags, e.g. -mavx2
- Minimalistic solution building only a subset of packages (see next)
- In use for GeantV studies
- Example of **experimental** selected microprocessor instruction sets support:
  - X86\_64+avx2+fma-centos7-gcc7-opt, x86\_64+avx2+fma-slc6-gcc62-opt,  
x86\_64+sse3-centos7-gcc62-opt, x86\_64+sse3-slc6-gcc7-opt,  
x86\_64+sse3-ubuntu18-gcc7-opt
- See [P Mato at HSF Packaging Forum](#), May 2018

# Support Multiple Micro-architectures (2)

- \* a complete 'view' is made using a mix of packages from different platform tags



# Evaluation of Spack technology

- Ongoing effort - to try out “Python” Spack instead of CMake “LCGCMake” stack build system
- First results and progress were reported by Ivan Razumov at HSF Packaging Working Group meeting:  
<https://indico.cern.ch/event/869507/contributions/3665788/>
- Subset of LCG release, alpha version of spack-based LCG release 97 for both Python 2.7 and 3.6 is built, buildcache is available and already used by FCC and Key4Hep projects (Valentin V).

Evaluation is not finished. We still need time to evaluate it.

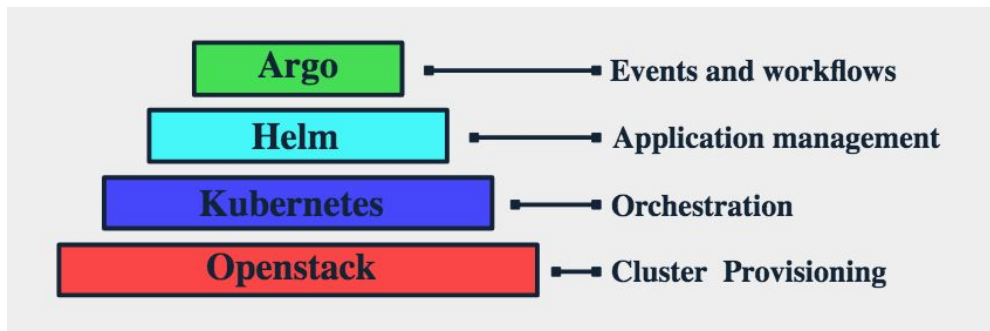
- Spack's concretizer: not correct behaviour with “virtual” packages, too “aggressive” hash algorithm - to be fixed in June (TBC)
- Recipes for some packages to be implemented in SPACK.

# K8s for LCG Stack (Richard Bachmann and Johannes Heinz)

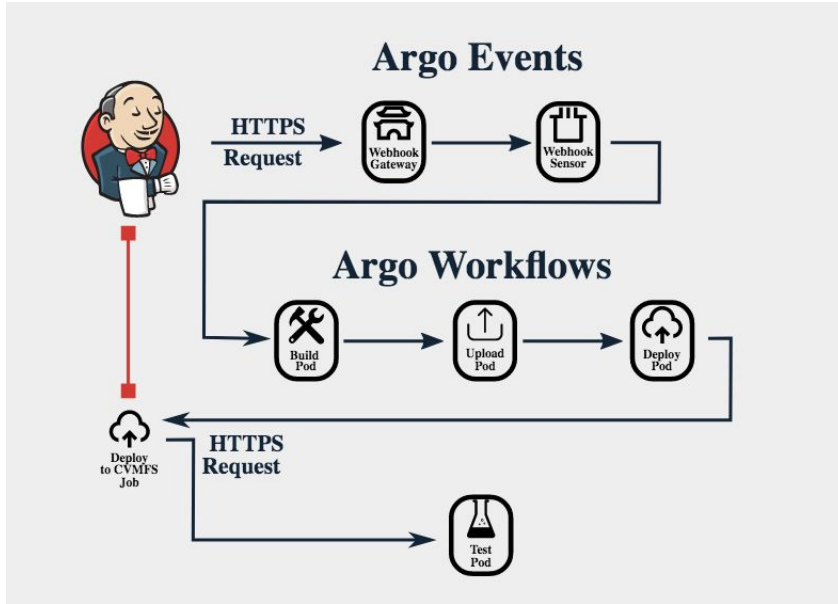
Motivation:

- exercise new promising technology
- optimize usage of resources (releasing/deleting OpenStack VMs)

current stack for the SPI Kubernetes setup:



# Current K8s-Argo system overview



- Jenkins - controller for builds
- Argo Events exposes webhooks for Jenkins
- Argo Workflows run each build stage;
- output sent back to Jenkins for CVMFS publication
- final result is tested with another container

More detailed talk will be given by Richard B. at EP-SFT group meeting soon.

Stay tuned!

# Summary

- Role of LCG stacks is to bridge experiments/user needs to OS
  - Driven by experiment/user requirements and feedback
- CernVM-FS main deployment area
  - Alternatively: tarfiles, RPMs, containers, ...
- Build/packaging tool
  - CMake-based in-house tool covers the needs
    - Including support for micro-architecture instruction sets
  - Eager to consider replacement when production grade
    - Spack being evaluated, not yet ready
- Prototype for build infrastructure based on K8s in progress
  - Solution to optimized use of resources
  - Possible derived improvements for base build infrastructure

# Event description

In this meeting, we will look at both the infrastructure initiatives on container distribution and librarian efforts to use modern packaging approaches. The related technologies are CernVM-FS, Spack, containers (docker, k8s, singularity).

The aim is to collect a status quo of technologies used in WLCG and to gain a better common understanding of what each of the technologies can do for us and how they fit in the picture of the overall goal.

Particular questions that can be addressed in the discussion are:

- What do we expect as an experiment software platform? For instance, the hep-os-libs meta-package, the CernVM virtual machine, or experiment base containers can all be seen as a platform.
- How are software stacks for multiple architectures managed, e.g. different AVX flavors, AArch64, etc.
- How do we expect sites to evolve: Is singularity here to stay? Will we see a significant fraction of kubernetes clusters?
- How do developers prefer to get access to the latest experiment software stacks?
- How are user containers on batch/grid resources supported?
- What are the experiment plans for software archival (preservation)
- Will future outreach efforts (master classes etc) use local software or do we expect them to be fully web based?